Amendments to the Specification

Please replace paragraph [0001] on pg. 1 with the following amended paragraph:

[0001] This application is related to the copending and commonly assigned United States patent application entitled "Method and Apparatus for Encoding Data to Guarantee Isolated Transitions in a Magnetic Recording System", having United States Application Serial No. 10/037,753 ~~attorney docket no. TUC920010037US1~~, which patent application was filed on January 2, 2005, the same date herewith and is incorporated herein by reference in its entirety.

Please replace paragraph [0030] on pgs. 10-11 with the following amended paragraph:

[0030] In the described implementations, the binary stream received by the PLL decode 42 includes the VFO field 101010 followed by the synchronization mark, and then by the data encoded using a 16/17 rate code, such as the rate code disclosed in the related patent application entitled "Method and Apparatus for Encoding Data to Guarantee Isolated Transitions in a Magnetic Recording System," having United States Application Serial No. 10/037,753 ~~attorney docket no. TUC920010037US1~~ and which patent application was incorporated herein by reference in its entirety above, wherein each codeword starts and ends with no more than three zeros and three ones, where two patterns 101010101 and 010101010 are excluded and each codeword contains a peak 010. To encode the EPR4 waveform, the binary stream is precoded by multiplying the binary stream by $1/(1 \oplus D)$, (where D is a delay operator) and then convolving the binary stream with $(1 - D)(1 + D2)$ to obtain the EPR4 waveform in a manner known in the art. To decode the EPR4 waveform, a Viterbi decoding algorithm is applied to a possibly noisy version of the waveform, and the result of the Viterbi algorithm is postcoded by convolving the result of the algorithm with $1 \oplus D$.

Please replace the paragraph at line 7 in paragraph [0013] on pg. 4 with the following amended paragraph:

FIG. 2 illustrates the recording or channel format of encoded data sets employing the synchronization and resynchronization marks in accordance with implementations of the invention; [[and]]

Please replace paragraph [0015] on pgs. 4-5 with the following amended paragraph:

[0015] FIG. 1 illustrates a tape storage environment in which aspects of the invention are implemented. A host system 2 is in communication with a tape drive 4. The host system 2 would include a storage interface or adaptor card (not shown), such as a Small Computer System Interface (SCSI), Fibre Channel, iScsi, etc., and the tape drive 4 would also include an adaptor card, where the adaptor cards enable communication between the devices. The [[host 2]] tape drive 4 would further include write data processing logic 6 comprising circuits which process data received from host 2 into data which will be written to tape. The write data processing logic 6 may include circuitry for data compression, Error Correction Code (ECC) encoding, and division of the data into blocks to be written to tape. Data from the write data processing logic 6 is then sent to a write formatter 8 that includes circuitry to perform run length limited (RLL) encoding of the data, add format fields as necessary, and include synchronization marks. Specifically, the RLL encoder 10 converts [[convert]] the processed data into an RLL encoded bit stream, the sync generator 12 generates the synchronization marks which are added to the data stream, and the other format fields 14 that generates all other format fields such as Data Set Separators (DSS), Voltage Controlled Oscillator (VCO) lock fields, headers, etc. Some format fields may be multiplexed into the data before the RLL encoder (e.g. the headers). A multiplexor (MUX) 16 is controlled to add the synchronization fields and other format fields to the RLL encoded bit stream as necessary per the logical format to be written to tape. The write signal processing 18 circuitry performs further processing on the RLL encoded bit stream, such as write equalization, etc. From there, the signal passes to write driver circuits 20 that drive the write heads 22 in response to a signal to be written. The write head 22 contains the actual write elements (e.g. inductive write elements). The data is written to a tape medium 24 within a tape cartridge 26.

Please replace paragraph [0018] on pg. 6 with the following amended paragraph:

[0018] In alternative implementations, the encoding and decoding logic may be implemented in other types of storage devices, such as a hard disk drive, optical disk drive or other device for reading and writing data to a non-volatile storage medium. In the described implementations, the tape medium [[8]] 24 comprises a magnetic tape (e.g., Linear Tape Open, Travan, etc.) or digital tape.

Please replace paragraph [0037] on pg. 14 with the following amended paragraph:

[0037] FIG. 5 illustrates how the sync detector 38 implements bit synchronization and resynchronization in accordance with implementations of the invention. The logic begins at block 300 where the read formatter 36 receives an encoded sequence of bits from the tape medium [[8]] 24. The sync detector 38 will seek to detect (at block 302) the VFO pattern in the encoded sequence of bits. Once the VFO is detected the sync detector 38 executes the logic of FIG. 3 to detect the synchronization mark S (at block 304). By identifying the synchronization mark (S) to correctly identify the beginning of the user data. While the sync detector 38 is reading the data (at block 306), the sync detector 38 seeks to identify any resynchronization marks (RS) to verify the data is being read correctly. The sync detector 38 would execute the logic of FIG. 4 (at block 308) to identify the resynchronization (RS) pattern and test whether data is properly interpreted.

Please replace paragraph [0020] on pg. 7 with the following amended paragraph:

[0020] In certain implementations, the input at the write head [[2]] 22 and the read head 30 are asynchronous. Thus, the data stream into the read channel must provide the clocking necessary to allow the read head 30 to correctly interpret the transitions of the stored data so that read formatter 36 may decode the channel output data.

Please replace paragraph [0022] on pgs. 7-8 with the following amended paragraph:

[0022] FIG. 2 illustrates the recording or channel format of encoded data sets employing the synchronization and resynchronization marks in accordance with implementations of the invention. Each data set is separated by a data set separator, or gap 50. When the data is recorded by a magnetic tape drive, the gap is typically called the "IBG" or interblock gap. The gap is sometimes followed by a variable frequency oscillator (VFO) area 52, which is utilized ~~by the output channel PLL 20~~ to [[maintain]] synchronize the data frequency to that of the VFO area. In data recording, the tape drive 4 may operate at various speeds, or may stop and restart or reverse direction, therefore causing changes in the data frequency, which must be determined by detecting the VFO area frequency. The VFO area 52 is then followed by the encoded data 53-55 quadrants, and, in tape drives that may move forward or backwards, such as magnetic tape, a backwards VFO area 57.

Please replace paragraph [0023] on pg. 8 with the following amended paragraph:

[0023] In the described implementations, a grouping of codeword groups, or codeword quad 54, is expanded and shown in greater detail in FIG. 2. Each codeword quad includes a header with a leading VFO pattern 60 and a synchronization mark 61, a plurality of codeword groups 62-63, separated by a resynchronization mark 65, and having a "Reverse Sync" pattern 68 for magnetic tape, at the end of the quad, adjacent to the next VFO pattern. Each of the codeword groups 62-63 may be of any suitable length. It is also preferable that the VFO pattern 60 be encountered before the synchronization mark 61 and concatenated therewith, so that the frequency may be adjusted before the decoder [[40]] is aligned to the synchronization mark. The described implementations may be configured to read and write in both directions by writing the synchronization mark [[68]] 61 in the opposite side of the VFO pattern and in reverse order, and concatenated with the trailing VFO pattern. Thus, the reverse sync 68, the VFO 60 and the next synchronization mark 61 may be concatenated in a sequential string.

Please replace paragraph [0026] on pgs. 8-9 with the following amended paragraph:

[0026] Errors in the VFO and/or synchronizing patterns may result in finding the synchronizing pattern in the wrong place, with the consequence that the recording read channel begins supplying "data" before or after the data actually starts. This would cause the [[PLL]] RLL decode 42 to operate out of synchronization, thereby preventing any of the supplied "data" from being decoded. One error occurring in data recording systems is a "bit shift," where the inter-symbol interference of high frequency signals shifts the detection of any transition away from an adjacent transition and towards a long string of no transitions (many "0s" between the transitions). This causes the detection circuitry to shift the transitions of longer spans into different positions such that they may instead be erroneously detected as shorter patterns.

Please replace paragraph [0028] on pg. 9 with the following amended paragraph:

[0028] In the described implementations, an error detection algorithm based on runs of consecutive errors between peaks of a synchronization mark is disclosed which would prevent the errors discussed above. The algorithm according to the described implementations utilizes a synchronization mark consisting of isolated peaks. Isolated peaks are used because the peaks act as natural barriers against the most likely type of error propagation found in decoding EPR4

("Extended Partial Response") type of waveforms. In one implementation using a 16/17 encoding rate, a 17-bit synchronization symbol S may be used, such as S = 01000100001010001, which is appended to the VFO. Alternative synchronization symbols may be generated into the data stream. Generating synchronization symbols (S) before the data in the data stream enables the [[PLL]] RLL decode 42 to detect the point at which data decoding begins in the data stream. In certain implementations, the synchronization symbol is appended to the VFO to obtain:

S' = 1001000100001010001.

Because the first two bits (10) of S' are already in the VFO, seventeen bits, instead of 19, are appended to the VFO, which is equivalent to two bytes in a 16/17 code rate . In this way, the propagation peak is exploited to incorporate the synchronization symbol into a propagation peak. The value of the synchronization symbol (S') can be generated in alternative manners. After S' has been determined, the data can be located and read because the data begins following the synchronization mark. In addition, since the data may also be read in the reverse direction, the synchronization mark is also recorded in the reverse direction after the encoded data.. As a result, no portion of the synchronization mark is likely to be confused with the VFO pattern, and the synchronization mark is unlikely to be recognized until the full pattern is actually encountered. Thus, if a valid synchronization mark is encountered before encoded data is, it will likely be recognized in the correct location.

Please replace paragraph [0030] on pgs. 10-11 with the following amended paragraph:

[0030] In the described implementations, the binary stream received by the [[PLL]] RLL decode 42 includes the VFO field 101010 followed by the synchronization mark, and then by the data encoded using a 16/17 rate code, such as the rate code disclosed in the related patent application entitled "Method and Apparatus for Encoding Data to Guarantee Isolated Transitions in a Magnetic Recording System," having attorney docket no. TUC920010037US1 and which patent application was incorporated herein by reference in its entirety above, wherein each codeword starts and ends with no more than three zeros and three ones, where two patterns 101010101 and 010101010 are excluded and each codeword contains a peak 010. To encode the EPR4 waveform, the binary stream is precoded by multiplying the binary stream by $1/(1 \oplus D)$, (where D is a delay operator) and then convolving the binary stream with $(1 - D)(1 + D2)$ to obtain the EPR4 waveform in a manner known in the art. To decode the EPR4 waveform, a

Viterbi decoding algorithm is applied to a possibly noisy version of the waveform, and the result of the Viterbi algorithm is postcoded by convolving the result of the algorithm with $1 \oplus D$.